

Техническое задание: Система обмена сообщениями с функционалом жалоб и админ-панелью

1. Введение

Разработать веб-приложение для обмена сообщениями с возможностью подачи жалоб на контент или пользователей, адаптивным интерфейсом и админ-панелью.

Цель: Проверка навыков работы с PHP, MySQL, ролевой моделью, обработкой пользовательских отчетов и адаптивной версткой.

2. Функциональные требования

2.1. Общие требования

Адаптивный интерфейс, поддерживающий мобильные устройства (минимальное разрешение — 320px).

Навигационное меню с учетом роли пользователя (обычный пользователь/администратор).

2.2. Регистрация и авторизация

Регистрация с полями: логин, email, пароль (6+ символов), имя (опционально).

Роли:

- Пользователь — может отправлять сообщения, жаловаться на контент.
- Администратор — доступ к админ-панели, управление жалобами, пользователями.

2.3. Функционал пользователя

Отправка сообщений:

- Выбор получателя из списка.
- Максимальная длина сообщения — 500 символов.

Просмотр сообщений:

- Входящие/исходящие сообщения с сортировкой по дате.
- Возможность удалить свои сообщения (мягкое удаление).

Жалобы:

- Подача жалобы на сообщение или пользователя:
- Выбор типа жалобы (спам, оскорбление, нежелательный контент).
- Текстовое поле для описания (до 200 символов).

- Просмотр статуса своих жалоб (в личном кабинете).

2.4. Функционал администратора

Управление жалобами:

- Просмотр списка всех жалоб с фильтрами (тип, статус, дата).
- Изменение статуса жалобы ("новый", "в обработке", "решено").
- Принятие мер:
 - Удаление сообщения.
 - Блокировка пользователя.
 - Отправка уведомления нарушителю.

Админ-панель:

- Управление пользователями (блокировка, сброс пароля).
- Просмотр статистики: количество активных жалоб, популярные типы нарушений.

2.5. Дополнительные функции

Уведомления для администраторов о новых жалобах.

Ограничение на количество жалоб от одного пользователя (не более 5 в сутки).

3. Требования к базе данных

3.1. Таблицы

1. ****users****:

- id (INT, PK)
- login (VARCHAR(50), UNIQUE)
- password_hash (VARCHAR(255))
- email (VARCHAR(100), UNIQUE)
- role (ENUM('user', 'admin'))
- is_blocked (BOOLEAN DEFAULT 0)
- created_at (DATETIME)

2. ****messages****:

- id (INT, PK)
- sender_id (INT, FOREIGN KEY)

- receiver_id (INT, FOREIGN KEY)
- message_text (TEXT)
- is_deleted (BOOLEAN DEFAULT 0)
- sent_at (DATETIME)

3. ****reports**** (жалобы):

- id (INT, PK)
- user_id (INT, FOREIGN KEY) — автор жалобы.
- target_type (ENUM('message', 'user')) — тип объекта жалобы.
- target_id (INT) — ID сообщения или пользователя.
- type (VARCHAR(50)) — тип нарушения (спам, оскорбление и т.д.).
- description (TEXT) — описание жалобы.
- status (ENUM('new', 'in_progress', 'resolved'))
- created_at (DATETIME)

4. **Безопасность**

Валидация прав доступа:

- Пользователь не может жаловаться на самого себя.
- Администратор не может удалять/блокировать других администраторов.

Защита от массовых жалоб:

- Проверка лимита (5 жалоб/сутки) через COUNT() в SQL.

5. **Критерии оценки**

1. Работоспособность: регистрация, авторизация, отправка/просмотр сообщений.
2. Корректность структуры БД.
3. Безопасность: хеширование паролей, защита от SQL-инъекций, нет доступа к админ-панели для пользователей без роли admin, корректная проверка лимита жалоб.
4. Подача жалобы, отображение статуса, обработка администратором.
5. Блокировка пользователя/удаление сообщения по результатам жалобы.
6. Адаптивность: форма жалобы и таблицы отображаются без ошибок на мобильных устройствах.
7. Дополнительно: реализация уведомлений для администраторов.

Примечание:

- Решение должно включать SQL-дамп БД и пример тестовых данных (3 пользователя, 10 сообщений, 5 жалоб).
- Код должен быть разделен на логические модули (регистрация, авторизация, жалобы, админ-панель).